

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method of managing a plurality of software development environments coupled to one another through a cross development environment, the method comprising:

receiving a first problem report from a first problem tracking tool in a first software development environment among the plurality of software development environments, wherein the first problem report has a first format associated with the first problem tracking tool;

in response to receiving the first problem report, converting the first problem report into a second format associated with a second problem tracking tool in a second software development environment among the plurality of software development environments using a mapping configured to convert problem reports in the first format into the second format, wherein the mapping is defined in a mapping data structure comprising a plurality of mapping entries, wherein at least one mapping entry includes a wildcarded field, and wherein the first problem report is converted into the second format using the wildcarded field;

forwarding the converted first problem report in the second format to the second problem tracking tool;

after forwarding the converted first problem report to the second problem tracking tool, detecting an update made in a the first software development environment among the plurality of software development environments, wherein the update made in the first software development environment includes a change to the first format; and

dynamically modifying a the mapping between the first software development environment and a second software development environment among the plurality of software development environments in response to the detected update;

receiving a second problem report from the first problem tracking tool, the second problem report formatted in the changed first format;
in response to receiving the second problem report, converting the second problem report into the second format using the modified mapping; and
forwarding the converted second problem report in the second format to the second problem tracking tool.

2.-3. (Canceled).

4. (Original) The method of claim 1, wherein detecting the update comprises receiving a notification from the first software development environment.

5. (Original) The method of claim 1, further comprising notifying an administrator of the cross development environment in response to detecting the update.

6. (Original) The method of claim 1, further comprising notifying an administrator of the second software development environment in response to detecting the update.

7. (Original) The method of claim 6, further comprising updating the second software development environment in response to notification of the administrator.

8. (Original) The method of claim 1, wherein the update comprises an update to content stored in the first software development environment.

9. (Original) The method of claim 8, wherein the update comprises an update to at least one of a product, component and release stored in a library repository in the first software development environment.

10. (Original) The method of claim 1, wherein the update comprises an update to at least one of a tool, a parameter and a value in the first software development environment.

11. (Canceled).

12. (Original) The method of claim 1, further comprising transforming a transaction generated by the first software development environment into a format compatible with the second software development environment using the mapping.

13. (Original) The method of claim 12, wherein transforming the transaction includes routing the transaction to one of a plurality of cross development environment processes.

14. (Original) The method of claim 13, wherein routing the transaction is performed by a router process, the router process configured to perform at least one of failover and load balancing in connection with routing the transaction to a cross development environment process.

15. (Original) The method of claim 12, wherein transforming the transaction includes communicating the transaction to the second software development environment, the method further comprising retrying communication of the transaction to the second software development environment in response to unavailability of the second software development environment.

16.-18. (Canceled).

19. (Currently Amended) An apparatus, comprising:

a memory configured to store a mapping data structure for use in a cross development environment that couples together a plurality of software development environments, wherein the mapping data structure includes a plurality of mapping entries, wherein at least one mapping entry includes a wildcarded field;

a processor; and

program code configured to;

receive a first problem report from a first problem tracking tool in a first software development environment among the plurality of software development environments, wherein the first problem report has a first format associated with the first problem tracking tool;

in response to receiving the first problem report, convert the first problem report into a second format associated with a second problem tracking tool in a second software development environment among the plurality of software development environments using the mapping data structure, wherein the mapping data structure is configured to convert problem reports in the first format into the second format, and wherein the first problem report is converted into the second format using the wildcarded field;

forward the converted first problem report in the second format to the second problem tracking tool;

after forwarding the converted first problem report to the second problem tracking tool, detect an update made in a the first software development environment among the plurality of software development environments, wherein the update made in the first software development environment includes a change to the first format; and

dynamically modify the mapping data structure to modify a mapping between the first software development environment and a second software development environment among the plurality of software development environments in response to the detected update;

receive a second problem report from the first problem tracking tool, the second problem report formatted in the changed first format;

in response to receiving the second problem report, convert the second problem report into the second format using the modified mapping data structure; and

forward the converted second problem report in the second format to the second problem tracking tool.

20.-21. (Canceled).

22. (Original) The apparatus of claim 19, wherein the program code is configured to detect the update by receiving a notification from the first software development environment.

23. (Original) The apparatus of claim 19, wherein the program code is further configured to notify an administrator of the cross development environment in response to detecting the update.

24. (Original) The apparatus of claim 19, wherein the program code is further configured to notify an administrator of the second software development environment in response to detecting the update.

25. (Original) The apparatus of claim 19, wherein the update comprises an update to content stored in the first software development environment.

26. (Original) The apparatus of claim 25, wherein the update comprises an update to at least one of a product, component and release stored in a library repository in the first software development environment.

27. (Original) The apparatus of claim 19, wherein the update comprises an update to at least one of a tool, a parameter and a value in the first software development environment.

28. (Canceled).

29. (Original) The apparatus of claim 19, wherein the program code is further configured to transform a transaction generated by the first software development environment into a format compatible with the second software development environment using the mapping data structure.

30. (Original) The apparatus of claim 29, further comprising a plurality of cross development environment processes and a router process configured to receive the transaction and route the transaction to one of the cross development environment processes.

31. (Original) The apparatus of claim 30, wherein the router process is configured to perform at least one of failover and load balancing in connection with routing the transaction to a cross development environment process.

32. (Original) The apparatus of claim 29, wherein the program code is further configured to communicate the transformed transaction to the second software development environment, and to retry communication of the transaction to the second software development environment in response to unavailability of the second software development environment.

33.-38. (Canceled).